Introduction to Image Analysis
Feature Extraction

Mauricio Reyes
mauricio.reyes@unibe.ch

University of Bern
ARTORG Center for Biomedical Engineering Research
Medical Image Analysis Group

Feature Extraction – Why?

- Raw pixels are high-dimensional and sensitive to noise, illumination, and geometric transforms
- **Feature extraction** = transform raw image data into a compact, informative representation
- Good features are:
  - **Discriminative –** different structures yield different features
  - **Invariant –** stable under rotation, scale, illumination
  - **Compact –** low-dimensional, efficient to store and match

Feature Extraction – Overview

Keypoint-based (local)
SIFT, ORB

Texture-based (regional/global)
LBP, Gabor, GLCM

Shape-based
Fourier, Moments, Curvature

Radiomics (medical imaging)
$1^{st}/2^{nd}$ order, Shape, Filtered

This lecture covers the methods above, from classical computer vision features to their application in medical image analysis via **radiomics**.

SIFT

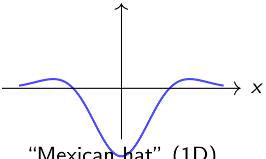
Scale-Invariant Feature Transform

Laplacian of Gaussian (LoG) – Detecting Blobs & Edges

Goal: detect regions of rapid intensity change (edges, blobs) at a **chosen scale**. Two steps combined in one filter:

- **Gaussian** – smooth the image to select a scale (controlled by σ)
- **Laplacian** – second derivative to find where intensity changes rapidly

$$\mathrm{LoG}(x, y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



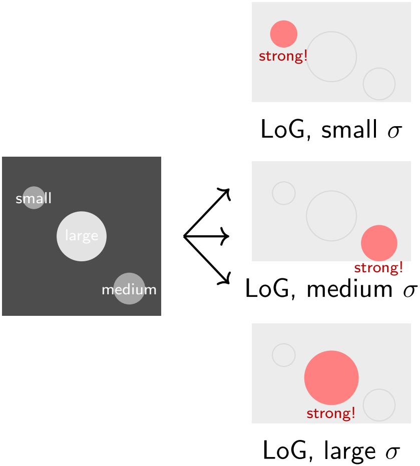
"Mexican hat" (1D)

The role of σ:

- Small $\sigma \Rightarrow$ responds to fine detail (small blobs, thin edges)
- Large $\sigma \Rightarrow$ responds to coarse structures (large blobs, broad edges)

By applying LoG at **multiple σ values**, we can detect features at different scales in the same image.

LoG – Visual Example



Each blob produces the strongest LoG response at the σ that **matches its size**. This is the foundation of multi-scale feature detection.

SIFT – Overview

Lowe, 2004

Goal: Detect and describe local image features that are invariant to scale, rotation, and partially to illumination changes.

Four steps:

1. **Scale-space extrema detection** – find candidate keypoints
2. **Keypoint localization** – refine and filter
3. **Orientation assignment** – for rotation invariance
4. **Keypoint descriptor** – a 128-dimensional fingerprint

SIFT – Building the Scale Space

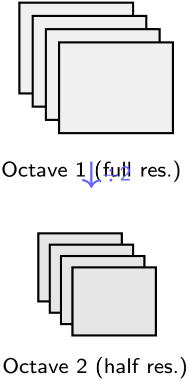
> **Gaussian scale space:** blur the image at increasing scales

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

> Images are organized in **octaves**:

— An octave = a group of images at the same resolution, blurred with increasing σ

— Between octaves, the image is **downsampled by factor 2**



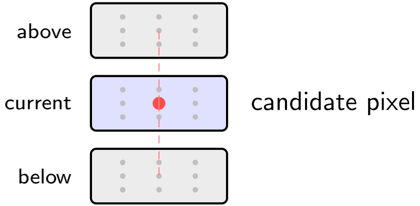
Within one octave, consecutive scales differ by factor $k = 2^{1/s}$

SIFT – Difference of Gaussians (DoG)

- To find keypoints, we need to detect “blobs” at different scales
- The **Difference of Gaussians** approximates the Laplacian of Gaussian (a blob detector), but is cheaper to compute:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

  Simply subtract adjacent blurred images within each octave.
- **Keypoint candidates:** pixels that are local extrema (maxima or minima) compared to all 26 neighbors in a $3 \times 3 \times 3$ cube (8 spatial + 9 above + 9 below in scale):



Compare with **26 neighbors** in space and scale

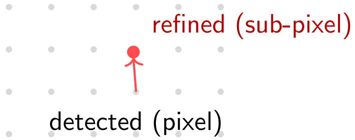
SIFT – Sub-Pixel Refinement

Problem: DoG extrema land on discrete pixel locations, but the true peak may lie **between pixels**.

- **Solution:** Fit a quadratic (Taylor expansion) around the detected extremum to find the precise peak:

$$D(\mathbf{x}) \approx D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

  Setting the derivative to zero gives an offset $\hat{\mathbf{x}}$ from the pixel grid.
- This yields **sub-pixel accurate** keypoint position and scale.



SIFT – Filtering Unstable Keypoints

Not all detected extrema are good keypoints. We discard two types:

1. **Low-contrast responses:**
   - If $|D(\hat{\mathbf{x}})| < 0.03$, the response is too weak $\Rightarrow$ reject
2. **Edge responses** (poorly localized along edges):
   - Use the **Hessian matrix H** – the 2×2 matrix of second derivatives (same concept as in the Harris corner detector):

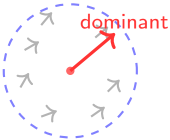
$$\mathbf{H} = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$$

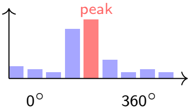
   - Its eigenvalues measure curvature in two perpendicular directions
   - **Edge:** one eigenvalue much larger than the other (strong curvature in one direction only)
   - Reject if: $\dfrac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} > \dfrac{(r+1)^2}{r}$ with $r = 10$

SIFT – Orientation Assignment

Each keypoint now has a position (x, y) and scale σ. We assign an **orientation** for rotation invariance.

- Compute gradient direction for every pixel in a window around the keypoint
- Build a **36-bin histogram** of directions ($360° \div 10° = 36$ bins), weighted by gradient magnitude
- The **highest peak** = keypoint orientation





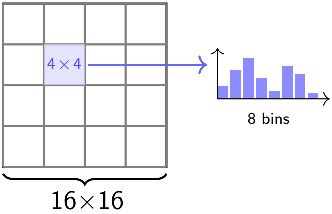
Why rotation-invariant?

All subsequent computations are done **relative to this direction**. If the image rotates, the peak rotates too, and the descriptor stays the same.

SIFT – Keypoint Descriptor (128-dim)

Build a compact “fingerprint” for each keypoint:

1. Take a 16×16 pixel patch, **rotated to align with the dominant orientation**
2. Divide into $4 \times 4 = 16$ sub-blocks (each 4×4 pixels)
3. In each sub-block, compute an **8-bin** gradient orientation histogram
4. Concatenate: $16 \times 8 =$ **128**-dim vector
5. Normalize (illumination robustness)



$16 \times 8 =$ **128** dimensions

Why sub-blocks?
Each sub-block captures gradients in its own spatial region. The descriptor “knows” that certain gradients are top-left vs. bottom-right of the keypoint.

SIFT – Strengths & Limitations

Strengths

- Scale-invariant
- Rotation-invariant
- Partially illumination-invariant
- Highly distinctive (128-dim)
- Excellent for matching and recognition

Limitations

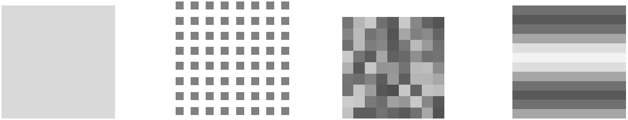
- Computationally expensive
- Not real-time (without GPU)
- Sparse features only

Faster alternatives: ORB (binary descriptor, very fast), AKAZE (nonlinear scale space)

TEXTURE FEATURES

Texture – What Is It?

Texture describes the spatial arrangement of intensities in a region – the visual “pattern” of a surface.



Smooth Regular Random Oriented

Key question: How do we *quantify* texture so a computer can distinguish these patterns?

Three approaches we will cover:

- **LBP** – encode local micro-patterns (pixel-level comparisons)
- **Gabor filters** – capture frequency and orientation content
- **GLCM** – model spatial co-occurrence of intensity pairs

LBP – Local Binary Patterns: The Idea

Ojala, Pietikäinen & Harwood, 1996

Idea: For each pixel, compare its intensity to its neighbors.
Encode the result as a binary number.

Step 1: Neighborhood

52	48	61
43	55	72
39	58	66

Center $g_c = 55$

$\longrightarrow$

Step 2: Threshold

0	**0**	**1**
0		**1**
0	**1**	**1**

1 if ≥ 55, else 0

$\longrightarrow$

Step 3: Encode

2^0	2^1	2^2
2^7		2^3
2^6	2^5	2^4

Bit positions
(clockwise from top-left)

LBP – Computing the Code

From the previous example:

Reading clockwise from top-left neighbor:

Neighbor	52	48	61	72	66	58	39	43
≥ 55?	0	0	1	1	1	1	0	0
Bit weight	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7

$\text{LBP} = 0 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 + 1 \cdot 16 + 1 \cdot 32 + 0 \cdot 64 + 0 \cdot 128 = \mathbf{60}$

General formula for P neighbors on a circle of radius R:

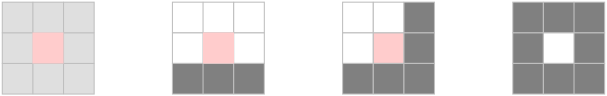
$$\text{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p, \qquad s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

For $P = 8$: LBP values range from 0 to 255.

Feature: Histogram of LBP values over a region $\Rightarrow$ texture descriptor.

LBP – What Do the Codes Represent?

Different LBP codes capture different local micro-patterns:



Flat area	Horizontal edge	Corner	Bright spot
LBP $\approx$ 0 or 255	e.g. LBP = 7	e.g. LBP = 31	LBP = 0

Uniform LBP: Patterns with ≤ 2 bit transitions ($0 \to 1$ or $1 \to 0$) cover $\sim$90% of all patterns in natural textures. Reduces histogram from 256 to $\sim$59 bins.

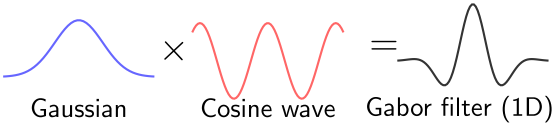
Advantages: very fast, invariant to monotonic gray-level changes, no training needed

Limitations: sensitive to noise, not inherently rotation- or scale-invariant

Gabor Filters – Intuition

Goal: Measure how much a local region matches a pattern at a specific **orientation** and **frequency** (scale).

- A Gabor filter = a **sinusoidal wave** (captures frequency) modulated by a **Gaussian envelope** (localizes in space)
- Think of it as asking: "Is there a stripe-like pattern here, in this direction, at this spacing?"



Biologically motivated: similar to receptive fields of neurons in the primary visual cortex (V1).

Gabor Filters – The 2D Filter

The 2D Gabor filter:

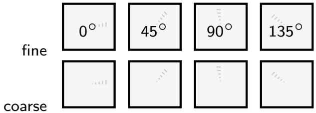
$$g(x, y) = \exp\Big(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\Big) \; \cos\Big(\frac{2\pi x'}{\lambda}\Big)$$

$$x' = x\cos\theta + y\sin\theta, \qquad y' = -x\sin\theta + y\cos\theta$$

Parameters:

- λ – wavelength (spacing between stripes)
- θ – orientation of the stripes
- σ – size of the Gaussian window
- γ – aspect ratio (elongation)

Usage: build a **filter bank** by varying θ and λ:



4 orientations $\times$ 2 scales $=$ 8 filters

Gabor Filters – Application & Properties

How to use:

> Convolve the image with each filter in the bank
> High response = the local texture matches that orientation and frequency
> **Feature vector:** mean and variance of each filter response over a region

Advantages

> Jointly localizes in space and frequency
> Tunable for specific textures
> Excellent for oriented structures (vessels, fibers)

Limitations

> Many parameters to tune
> Computationally heavier than LBP
> No optimal filter selection rule

Applications: texture segmentation, face recognition, fingerprint analysis, retinal vessel detection.

GLCM – Idea: Co-occurrence of Intensities

Haralick et al., 1973

LBP compares a pixel to its neighbors. GLCM asks a different question:

“How often does intensity i appear next to intensity j?”

- Choose a **distance** d and **angle** θ (e.g., $d = 1$, $\theta = 0^\circ =$ horizontal neighbor)
- Count how many times each pair (i, j) occurs $\Rightarrow$ **Gray-Level Co-occurrence Matrix**

Smooth image — GLCM: values on diagonal ($i \approx j$)

Rough image — GLCM: values spread everywhere ($i \neq j$)

GLCM – How to Compute

Example: 4-level image, $d = 1$, $\theta = 0^\circ$ (horizontal)

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Image (4 gray levels)

GLCM ($d = 1$, $\theta = 0^\circ$):

$i \backslash j$	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

E.g., the pair $(0, 1)$ appears 2 times horizontally. Normalize by total count $\Rightarrow P(i, j)$.

GLCM – Texture Features

From the normalized GLCM $P(i,j)$, we compute texture descriptors:

Contrast – measures local intensity variation:

$$\text{Contrast} = \sum_{i,j} (i-j)^2 \, P(i,j)$$

Homogeneity – high when GLCM values are near the diagonal:

$$\text{Homogeneity} = \sum_{i,j} \frac{P(i,j)}{1+|i-j|}$$

Energy – uniformity of the texture (high for constant regions):

$$\text{Energy} = \sum_{i,j} P(i,j)^2$$

Correlation – linear dependency between neighboring intensities:

$$\text{Correlation} = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)\, P(i,j)}{\sigma_i \, \sigma_j}$$

GLCM – Interpreting the Features

Feature	Smooth texture	Rough texture
Contrast	Low	High
Homogeneity	High	Low
Energy	High	Low
Correlation	High	Low

In practice:

- Compute GLCM for multiple distances d and angles θ
- Average the features across angles for rotation robustness
- GLCM features are widely used in medical imaging (see radiomics section)

SHAPE-BASED FEATURES

Shape-Based Features

After segmentation, the **shape** of a region carries important information.

Boundary-based:

- **Fourier descriptors:** decompose the contour into frequency components. Low frequencies = overall shape; high = fine detail.
- **Curvature:** local bending of the contour

Region-based:

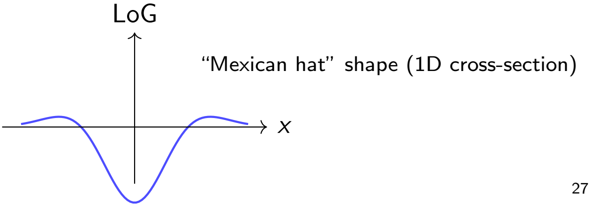
- **Compactness** $= \frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2}$ (1 for a circle, less for elongated shapes)
- **Eccentricity** $= \lambda_1/\lambda_2$ (ratio of principal axes of the inertia ellipse)
- **Hu moments:** 7 moment-based features invariant to translation, scale, and rotation
- **Sphericity** (3D) $= \frac{\pi^{1/3}(6V)^{2/3}}{A}$ (1 for a sphere)

Laplacian of Gaussian (LoG)

- The **Laplacian of Gaussian** (LoG) highlights regions of rapid intensity change (edges, blobs) at a specific scale

$$\mathrm{LoG}(x, y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- σ controls the **scale** of detected features: small $\sigma =$ fine detail; large $\sigma =$ coarse structures
- The DoG used in SIFT is an approximation of LoG
- In radiomics, LoG-filtered images are used to extract texture features at specific scales



Feature Methods – When to Use What

	SIFT	LBP	Gabor	GLCM	Shape
Type	Local keypoint	Texture	Texture	Texture	Region
Scale inv.	✓	–	Partial	–	–
Rotation inv.	✓	Variant*	Partial	Averaged	Partial†
Speed	Slow	Very fast	Medium	Fast	Fast
Best for	Matching	Classification	Oriented tex.	Co-occurrence	Geometry

*Rotation-invariant LBP variant exists. †Hu moments are rotation-invariant.

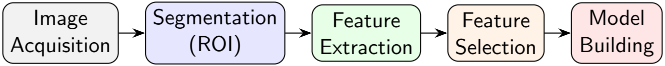
- **No single feature works best for all tasks**
- In practice: combine features or use learned features (CNNs)
- Medical imaging uses all of the above via **radiomics**

RADIOMICS

Feature Extraction for
Medical Imaging

Radiomics – What Is It?

Radiomics = high-throughput extraction of quantitative features from medical images (CT, MRI, PET) to uncover patterns not visible to the human eye.



- Coined in 2012 – combines feature extraction with machine learning for clinical decision support
- Key insight: "Images are more than pictures, they are data"
- Modalities: CT, MRI, PET – feature definitions are modality-independent

Radiomics – Feature Categories

Typically ∼100–1500 features per image region:

1. First-order (histogram) features – intensity statistics
   - Mean, std. dev., skewness, kurtosis, entropy
2. Shape-based features – geometry of the segmented region
   - Volume, surface area, sphericity, compactness, elongation
3. Texture features (2nd order) – spatial relationships
   - GLCM (as covered earlier), GLRLM, GLSZM, NGTDM
4. Filtered features – from transformed images
   - Apply LoG or wavelet filters first, then extract features from the filtered image

Radiomics – First-Order & Shape Features

First-order features (from the intensity histogram):

$$\text{Mean} = \frac{1}{N}\sum_{i=1}^{N} I_i, \qquad \text{Entropy} = -\sum_i p_i \log_2 p_i$$

$$\text{Skewness} = \frac{\frac{1}{N}\sum_i (I_i - \mu)^3}{\sigma^3}, \qquad \text{Kurtosis} = \frac{\frac{1}{N}\sum_i (I_i - \mu)^4}{\sigma^4}$$

Shape features (from the segmented ROI):

$$\text{Sphericity} = \frac{\pi^{1/3}(6V)^{2/3}}{A}$$

Other: elongation, flatness, max 2D/3D diameter, surface-to-volume ratio.

Radiomics – Texture Features

Beyond GLCM, radiomics uses additional texture matrices:

GLRLM (Gray-Level Run-Length Matrix):

- Counts consecutive pixels of the same intensity in a direction
- Features: short/long run emphasis, run-length non-uniformity
- Captures whether texture has long uniform stretches (smooth) or many short runs (heterogeneous)

GLSZM (Gray-Level Size Zone Matrix):

- Groups of *connected* pixels with the same intensity
- Captures size distribution of uniform zones

NGTDM (Neighborhood Gray-Tone Difference):

- Measures how much each gray level differs from its neighborhood average

All these features quantify different aspects of **tissue heterogeneity**.

Radiomics – Challenges & Best Practices

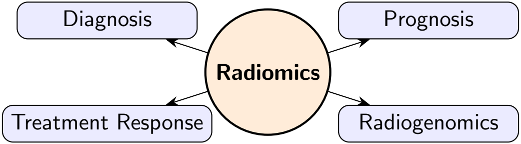
Challenges

- **Reproducibility:** features sensitive to scanner, protocol, segmentation
- **Overfitting:** 1000+ features, small patient cohorts
- **Redundancy:** many features are correlated
- Lack of standardization

Best Practices

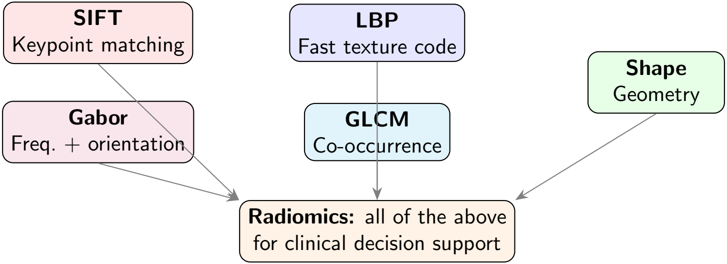
- Image normalization and resampling
- Standardized definitions (IBSI)
- Feature selection: LASSO, PCA, correlation filtering
- Cross-validation + external validation
- Tools: **PyRadiomics**, 3D Slicer

Radiomics – Clinical Applications



- **Oncology:** characterize lesion heterogeneity for diagnosis, predict survival, assess treatment response
- **Neurology:** Alzheimer's disease, stroke characterization
- **Cardiology:** myocardial tissue characterization
- Applicable to CT, MRI, PET – feature definitions are imaging-modality independent

Summary



- Choose features based on the task: matching ⇒ SIFT; texture classification ⇒ LBP/Gabor/GLCM; shape ⇒ moments
- Radiomics integrates all feature families for medical imaging
- Deep learning increasingly replaces handcrafted features, but understanding them remains essential

References

- D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” IJCV, 2004.
- T. Ojala et al., “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with LBP,” IEEE Trans. PAMI, 2002.
- R.M. Haralick et al., “Textural Features for Image Classification,” IEEE Trans. SMC, 1973.
- M.-K. Hu, “Visual Pattern Recognition by Moment Invariants,” IRE Trans. IT, 1962.
- R.J. Gillies et al., “Radiomics: Images Are More than Pictures, They Are Data,” Radiology, 2016.
- J.J.M. van Griethuysen et al., “Computational Radiomics System to Decode the Radiographic Phenotype” (PyRadiomics), Cancer Research, 2017.
- Image Biomarker Standardisation Initiative (IBSI), Radiology, 2020.